

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended): A method comprising the computer-implemented steps of:
detecting that a portion of a query execution plan to service a request for data will cause a first producer execution unit that will perform said portion, according to said query execution plan, to generate XML data for use by a second consumer execution unit in performing another portion of said query execution plan;
generating information to send to said first execution unit to cause said first execution unit to perform said portion of said query execution plan;
wherein said information would cause said first execution unit to generate said XML data in a first form that cannot be used by said second execution unit; and
annotating said information with an annotation that causes XML data generated by said first execution unit to be transformed to a canonical form for use by said second execution unit in performing said another portion of said query execution plan.
2. (Original): The method of Claim 1, wherein the step of generating information includes generating information that, prior to annotating said information, would cause said first execution unit to generate said XML data in a first form that cannot be used by said second execution unit, and wherein said canonical form is different from said first form.
3. (Original): The method of Claim 2, wherein said first form includes information to locate data that is stored in memory that is exclusive to said first execution unit, and wherein said information to locate data stored in said memory cannot be used by said second execution unit.
4. (Original): The method of Claim 1, wherein said request for data is a database query and said plan is a query plan.

5. (Original): The method of Claim 4, wherein said information is one or more database commands.
6. (Original): The method of Claim 1, wherein said annotation specifies a transformation operator.
7. (Original): The method of Claim 6, further comprising the computer-implemented steps of:
 - executing said transformation operator, by said first execution unit, to transform XML data generated by said first execution unit to said canonical form; and
 - sending XML data that is transformed by said first execution unit to said second execution unit in said canonical form.
8. (Original): The method of Claim 6, wherein said annotation specifies arguments for said transformation operator, to specify said canonical form.
9. (Original): The method of Claim 1, further comprising the computer-implemented steps of:
 - transforming, by said first execution unit, said XML data to said canonical form based on said annotation.
10. (Original): The method of Claim 1, wherein the step of annotating includes annotating said information with an operator to transform said XML data to a canonical form in which said XML data is serialized to represent particular data for a particular XML construct and is included in a serialized image that is sent to said second execution unit.
11. (Original): The method of Claim 1, wherein the step of annotating includes annotating said information with an operator to transform said XML data to a canonical form which includes an identifier of memory space where data is persistently stored, and wherein said data in said memory space is accessible by said second execution unit.
12. (Original): The method of Claim 1, wherein the step of annotating includes annotating said information with an operator to transform said XML data to a canonical form

in which said XML data is compressed according to a particular compression form that said second execution unit is able to decompress.

13. (Original): The method of Claim 1, wherein said first execution unit and said second execution unit are different execution units that are executing, in parallel, work associated with servicing said request.

14. (Original): The method of Claim 1, wherein said first execution unit and said second execution unit are different execution units that are each executing, on different servers of a distributed database system, work associated with servicing said request.

15. (Original): The method of Claim 1, wherein the steps of detecting, generating and annotating are performed by a means that distributes work associated with servicing said request to said first execution unit and said second execution unit, and wherein said first execution unit and said second execution unit are different execution units that are each executing work associated with servicing said request.

16. (Original): The method of Claim 15, wherein said first execution unit and said second execution unit are each executing, on different data sources, work associated with servicing said request.

17. (Original): The method of Claim 15, wherein said means that distributes work comprises an application server.

18. (Original): The method of Claim 15, wherein said means that distributes work comprises an application that manages workload among multiple means for executing said work.

19. (Original): The method of Claim 1, further comprising the computer-implemented steps of:

determining said canonical form from information that describes preferences of each of multiple execution units that performs work associated with servicing said request.

20. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.
21. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.
22. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.
23. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.
24. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.
25. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.
26. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.
27. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

28. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.
29. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.
30. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.
31. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.
32. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.
33. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 14.
34. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 15.
35. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 16.

36. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 17.
37. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 18.
38. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 19.
39. (currently amended): A method for processing XML data, comprising the computer-implemented steps of:
 - receiving information at a first execution unit to cause said first execution unit to perform work associated with a query execution plan for servicing a request for data; wherein said information comprises an annotation that causes the XML data generated by said first execution unit to be transformed to a canonical form for use by a second execution unit; wherein said information, without said annotation, would cause said second execution unit to receive from said first execution unit XML data in a first form that cannot be used by said second execution unit;
 - transforming XML data generated by said first execution unit to said canonical form prior to providing said XML data to said second execution unit; and
 - providing XML data that is transformed to said second execution unit in said canonical form for use in performing work associated with said query execution plan by said second execution unit.
40. (Original): The method of Claim 39, wherein the step of transforming said XML data to said canonical form is performed by said first execution unit.

41. (Original): The method of Claim 40, wherein the step of transforming comprises executing an operator specified in said annotation.
42. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 39.
43. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 40.
44. (currently amended): A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 41.
45. (Original): A database system comprising:
a query optimizer that receives a database query, formulates a query plan based on said query, and sends information based on said plan to a first execution unit;
wherein formulating a plan includes determining that said first execution unit produces XML data for use by a second execution unit, and determining whether said first execution unit produces said XML data in a first form that said second execution unit is able to use;
said first execution unit that receives said information from said query optimizer; and
said second execution unit that receives said XML data from said first execution unit.
46. (Original): The system of Claim 45,
wherein, if it is determined that said second execution unit is able to use said XML data in said first form,
said information that said query optimizer sends to said first execution unit comprises a direction to send said XML data in said first form to said second execution unit;

said first execution unit produces XML data in said first form while servicing said query, and sends said XML data to said second execution unit; and

 said second execution unit receives said XML data in said first form, and services said query based on said XML data in said first form.

47. (Original): The system of Claim 45,
wherein, if it is determined that said second execution unit is unable to use said XML data in said first form,
 said information that said query optimizer sends to said first execution unit comprises transformation information that causes said first execution unit to transform said XML data that is produced by said first execution unit to a second form that said second execution unit is able to use;
 said first execution unit produces transformed XML data in said second form based on said transformation information while servicing said query, and sends said transformed XML data to said second execution unit; and
 said second execution unit receives said transformed XML data in said second form, and services said query based on said transformed XML data.

48. (Original): The system of Claim 45, wherein said first execution unit and said second execution unit are different execution units that are servicing said request by performing work in parallel.

49. (Original): The system of Claim 45, wherein said first execution unit and said second execution unit are different execution units that are servicing said request by performing work on different servers of a distributed database system.

50. (currently amended): A system comprising:
 means for detecting that a portion of a query execution plan to service a request for data will cause a first producer execution unit that will perform said portion, according to said query execution plan, to generate XML data for use by a second consumer execution unit in performing another portion of said query execution plan;

means for generating information to send to said first execution unit to cause said first execution unit to perform said portion of said query execution plan;

wherein said information would cause said first execution unit to generate said XML data in a first form that cannot be used by said second execution unit; and

means for annotating said information with an annotation that causes XML data generated by said first execution unit to be transformed to a canonical form for use by said second execution unit in performing said another portion of said query execution plan.